Desenvolvimento de algoritmos sequenciais através de pseudocódigo (Ferramenta VisuAlg)

APRESENTAÇÃO

Nesta Unidade de Aprendizagem, estudaremos a solução de problemas através do desenvolvimento de algoritmos sequenciais em forma de pseudocódigo, além de acompanhar a análise e o estudo de várias aplicações práticas utilizando a ferramenta VisuAlg.

Bons estudos.

Ao final desta Unidade de Aprendizagem, você deve apresentar os seguintes aprendizados:

- Analisar algoritmos sequenciais em forma de pseudocódigo desenvolvidos na ferramenta VisuAlg.
- Demonstrar algoritmos sequenciais em pseudocódigo na ferramenta VisuAlg.
- Usar o pseudocódigo na representação da solução de problemas.

INFOGRÁFICO

A imagem descreve os temas que serão tratados nessa Unidade de Aprendizagem.

CONTEÚDO DO LIVRO

Um algoritmo em pseudocódigo completo é composto por declarações (variáveis, constantes, funções etc.), comandos (atribuição, entrada, saída) e estruturas que podem ser sequenciais, de seleção e de iteração (repetição) que permitem representar a solução de qualquer algoritmo. Os algoritmos que foram apresentados até o momento foram construídos apenas com a estrutura de controle sequencial, em que um grupo de comandos é executado linearmente, um após o outro.

Para melhor compreender a estrutura de controle sequencial em pseudocódigo, acompanhe a parte prática do capítulo de algoritmos sequenciais da seguinte obra: EDELWEISS, N.; LIVI, M.A.C. Algoritmos e programação com exemplos em Pascal e C - Vol. 23. Série Livros Didáticos Informática UFRGS. Porto Alegre: Bookman, 2014.

Boa leitura.

23

■■ série livros didáticos informática ufrgs ■■

algoritmos e programação

com exemplos em Pascal e C

■■ nina edelweiss

■■ maria aparecida castro livi

··→ as autoras

Nina Edelweiss é engenheira eletricista e doutora em Ciência da Computação pela Universidade Federal do Rio Grande do Sul. Durante muitos anos, lecionou em cursos de Engenharia e de Ciência da Computação na UFRGS, na UFSC e na PUCRS. Foi, ainda, orientadora do Programa de Pós-Graduação em Ciência da Computação da UFRGS. É coautora de três livros, tendo publicado diversos artigos em periódicos e em anais de congressos nacionais e internacionais. Participou de diversos projetos de pesquisa financiados por agências de fomento como CNPq e FAPERGS, desenvolvendo pesquisas nas áreas de bancos de dados e desenvolvimento de software.

Maria Aparecida Castro Livi é licenciada e bacharel em Letras, e mestre em Ciência da Computação pela Universidade Federal do Rio Grande do Sul. Desenvolveu sua carreira profissional na UFRGS, onde foi programadora e analista de sistema, antes de ingressar na carreira docente. Ministrou por vários anos a disciplina de Algoritmos e Programação para alunos dos cursos de Engenharia da Computação e Ciência da Computação. Sua área de interesse prioritário é o ensino de Linguagens de Programação, tanto de forma presencial quanto a distância.

E22a Edelweiss, Nina.
Algoritmos e programação com exemplos em Pascal e C [recurso eletrônico] / Nina Edelweiss, Maria Aparecida Castro Livi. – Dados eletrônicos. – Porto Alegre : Bookman, 2014.

Editado também como livro impresso em 2014.
ISBN 978-85-8260-190-7

1. Informática. 2. Algoritmos – Programação. I. Livi, Maria Aparecida Castro. II. Título.

CDU 004.421

Catalogação na publicação: Ana Paula M. Magnus – CRB 10/2052

Ainda utilizando as mesmas variáveis, os comandos a seguir são inválidos:

```
nome ← 10             { STRING NÃO PODE RECEBER VALOR NUMÉRICO }
letra ← i > 2         { CARACTERE NÃO PODE RECEBER VALOR LÓGICO }
letra ← nome          { VARIÁVEL CARACTERE NÃO PODE RECEBER STRING }
letra ← letra + 10  { EXPRESSÃO À DIREITA ESTÁ INCORRETA}
```

3.4 ⇢ fluxograma de programas sequenciais

No Capítulo 1 foram mostrados alguns blocos utilizados em fluxogramas, incluindo os que representam comandos de entrada, de saída e de atribuição (Figura 1.4). Diversas formas para representar entradas e saídas podem ser encontradas na literatura. Nos blocos adotados nesse livro é utilizado o mesmo bloco para ambas, identificando claramente a ação a ser executada (entrada ou saída). A Figura 3.1 mostra um fluxograma em que é feita uma entrada de dados que preenche a variável `valor`, sendo, em seguida, informado qual o valor lido.

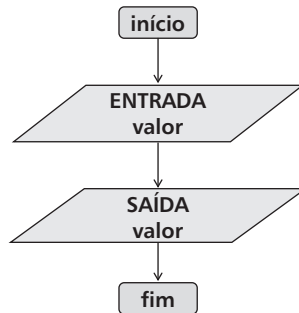



figura 3.1 Fluxograma com entrada e saída de dados.

Um programa pode ter vários comandos de entrada e de saída de dados em lugares diferentes. As formas dos blocos que representam esses comandos mostram visualmente, no fluxograma, os pontos de interação do programa com o usuário.

O comando de atribuição é representado através de um retângulo, dentro do qual é escrito o nome da variável, o símbolo que representa a atribuição (←) e a expressão, em sua forma matemática, ou seja, sem necessidade de representá-la em uma só linha. Como exemplo, a Figura 3.2 mostra o fluxograma que corresponde ao problema apresentado na Seção 1.2:

1. `obter os dois valores`
2. `realizar a soma`
3. `informar o resultado`

O primeiro passo corresponde a um comando de entrada de dados, em que são lidos dois valores. Para armazenar os valores lidos devem ser declaradas duas variáveis, `valor1` e `valor2`, de tipos compatíveis com os valores que serão fornecidos na entrada. No segundo passo, é

realizada a operação de soma dos valores contidos nas duas variáveis, sendo o resultado armazenado em outra variável chamada de `soma`. O terceiro passo corresponde a um comando de saída, através do qual o valor armazenado na variável `soma` é informado ao usuário. As setas indicam a sequência em que os comandos são executados.

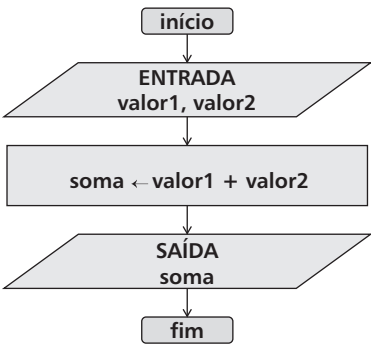



figura 3.2 Fluxograma da soma de dois valores.

3.5 ⇢ estrutura de um algoritmo

Nesta seção será montado o primeiro algoritmo completo utilizando as declarações e os comandos vistos até aqui. Será utilizado o mesmo exemplo da seção anterior (soma de dois valores), para o qual já foi construído o fluxograma.

Um algoritmo deve sempre iniciar com um **cabeçalho**, no qual o objetivo do algoritmo deve ser claramente identificado. A primeira linha desse cabeçalho deve trazer o nome do algoritmo, o qual, por si só, deve dar uma indicação das ações a serem executadas pelo mesmo. No caso do exemplo, o algoritmo foi chamado de `Soma2`, pois vai efetuar a soma de dois valores. Na linha seguinte do cabeçalho, na forma de um comentário, deve ser explicado o objetivo do algoritmo. Essa explicação é útil principalmente nos casos em que o nome do algoritmo não é suficientemente autoexplicativo. Cabeçalho do exemplo utilizado:

```
Algoritmo Soma2
{INFORMA A SOMA DE 2 VALORES LIDOS}
```

Logo após o cabeçalho vem a seção das **declarações** de variáveis, de constantes e de tipos. Para facilitar o entendimento de um algoritmo, é importante identificar claramente as `variáveis de entrada` e `de saída`, pois elas fazem a interface do usuário com o programa. As demais variáveis utilizadas durante o processamento, denominadas `variáveis auxiliares`, são declaradas em uma linha especial. Essa separação desaparece ao se traduzir o algoritmo para uma linguagem de programação, mas é aconselhável que seja acrescentada ao programa na forma de um comentário.

A declaração de variáveis do `Algoritmo Soma2` é a seguinte:

```
Entradas: valor1, valor2 (real)    {VALORES LIDOS}
Saídas: soma (real)
```

Os nomes escolhidos para as variáveis devem ser curtos e indicar qual a informação que elas irão armazenar. Caso isso não fique claro somente através do nome escolhido, é aconselhável escrever comentários explicando o significado de cada variável.

Após a seção de declarações, vem a área de **comandos**, delimitada pelas palavras reservadas `início` e `fim`. Cada comando deve ser escrito em uma linha separada. Ao contrário das linguagens de programação Pascal e C, a pseudolinguagem utilizada não emprega símbolo para separar comandos, sendo essa separação identificada somente pela posição de cada comando no algoritmo.

É importante utilizar **comentários** ao longo do algoritmo, indicando as ações que estão sendo executadas em cada passo. Isso auxilia muito os testes e a depuração do programa.

A estrutura básica de um algoritmo, com os elementos discutidos até o momento, é:

```
Algoritmo <nome do algoritmo>
{descrição do objetivo do algoritmo}
<declarações>
início
<comandos>
fim
```

Em declarações aparecem com frequência alguns ou todos os seguintes elementos:

```
Entradas: <lista de nomes de variáveis com seus tipos>
Saídas: <lista de nomes de variáveis com seus tipos>
Variáveis auxiliares: <lista de nomes de variáveis com seus tipos>
```

O algoritmo completo do exemplo da soma de dois valores é:

```
Algoritmo 3.1 – Soma2
{INFORMA A SOMA DE DOIS VALORES LIDOS}
  Entradas: valor1, valor2 (real){VALORES LIDOS}
  Saídas:   soma (real)
início
  ler (valor1, valor2)          {OBTÉM OS 2 VALORES}
  soma ← valor1 + valor2        {CALCULA A SOMA}
  escrever (soma)               {INFORMA A SOMA}
fim
```

Nos exercícios de fixação a seguir, recomenda-se definir inicialmente o(s) resultado(s) a produzir, a(s) entrada(s) a obter e, só então, tentar determinar um modo de solução. Procurar

identificar, nas soluções fornecidas, quais as linhas que correspondem, respectivamente, à entrada de dados, ao processamento e à apresentação dos resultados.

Observar que todos os problemas discutidos seguem o esquema básico destacado no início deste capítulo: entrada de dados, processamento e saída de dados.

3.6 ⇢ exercícios de fixação

exercício 3.1 Fazer um programa que recebe três notas de alunos e fornece, como saídas, as três notas lidas, sua soma e a média aritmética entre elas.

A Figura 3.3 mostra o fluxograma deste programa. Inicialmente são lidas as três notas, que são também impressas para que o usuário possa verificar o que foi lido. Em seguida, é calculada e informada a soma. Finalmente, é efetuado o cálculo da média, que é também informado ao usuário. A utilização de diversos comandos de saída neste programa permite ao programador verificar quais os valores intermediários do processamento, auxiliando a depurar o programa.

O algoritmo desse programa acrescenta as declarações das variáveis utilizadas, que não aparecem no fluxograma. São incluídos também comentários para explicar os diferentes passos do algoritmo.

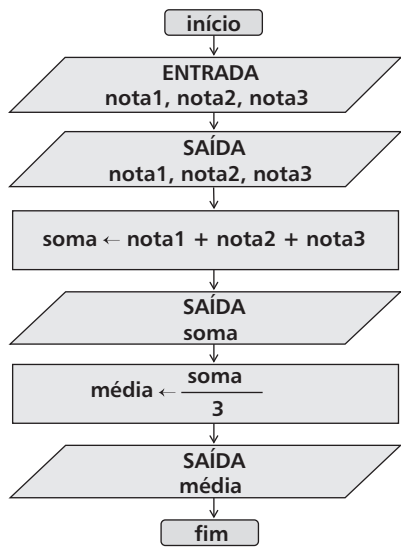



figura 3.3 Fluxograma do cálculo da média de três notas.

```
Algoritmo Média1
{INFORMA A SOMA E A MÉDIA DAS 3 NOTAS DE UM ALUNO}
  Entradas: nota1, nota2, nota3 (real)
  Saídas: soma, média (real)
início
  ler (nota1, nota2, nota3)          {ENTRADA DAS 3 NOTAS}
  escrever (nota1, nota2, nota3)     {INFORMA AS NOTAS LIDAS}
  soma ← nota1 + nota2 + nota3       {CALCULA A SOMA}
  escrever (soma)                    {INFORMA SOMA}
  média ← soma / 3                   {CALCULA A MÉDIA}
  escrever (média)                   {INFORMA MÉDIA CALCULADA}
fim
```

exercício 3.2 Dado o raio de um círculo, construir um algoritmo que calcule e informe seu perímetro e sua área.

Fórmulas: $\text{perímetro} = 2 \times \pi \times \text{raio}$

$\text{área} = \pi \times \text{raio}^2$

```
Algoritmo Círculo
{INFORMA O PERÍMETRO DE UM CÍRCULO E SUA ÁREA}
  Entrada: raio (real)
  Saídas: perímetro, área (real)
início
  ler (raio)                         {LÊ O RAIO DO CÍRCULO}
  perímetro ← 2 * 3,14 * raio        {CALCULA O PERÍMETRO}
  área ← 3,14 * sqr(raio)            {CALCULA A ÁREA}
  escrever (perímetro, área)         {INFORMA PERÍMETRO E ÁREA}
fim
```

exercício 3.3 Dado o preço de um produto em reais, converter este valor para o equivalente em dólares. O programa deverá ler do teclado o preço do produto e a taxa de conversão para o dólar.

```
Algoritmo ConversãoParaDólar
{CONVERTE UM VALOR EM REAIS PARA DÓLARES}
  Entradas: preço (real)    {PREÇO EM REAIS}
            taxa (real)     {TAXA DE CONVERSÃO PARA DÓLAR}
  Saída: emdolar (real)     {PREÇO EM DÓLARES}
início
  ler (preço)               {LÊ O PREÇO EM REAIS}
  ler (taxa)                {LÊ A TAXA DO DÓLAR}
  emdolar ← preço * taxa    {CALCULA O VALOR EM DÓLARES}
  escrever (emdolar)        {INFORMA O VALOR EM DÓLARES}
fim
```

exercício 3.4 Escrever um algoritmo que calcula a comissão de um vendedor sobre uma venda efetuada. O algoritmo deve ler o número de um vendedor, o valor da venda efetuada e o percentual a receber sobre a venda.

```
Algoritmo ComissãoSobreVenda
{CALCULA A COMISSÃO DE UM VENDEDOR SOBRE UMA VENDA}
  Entradas: numVendedor (inteiro)      {NÚMERO DO VENDEDOR}
            valorVenda (real)          {VALOR DA VENDA}
            percentual (real)          {PERCENTUAL A RECEBER}
  Saídas: comissão (real)              {COMISSÃO A RECEBER}
início
  ler (numVendedor, valorVenda)              {LEITURA DADOS VENDA}
  ler (percentual)                           {LEITURA PERCENTUAL}
  comissão ← valorVenda * percentual * 0,01 {CÁLCULO DA COMISSÃO}
  escrever (numVendedor, comissão)           {SAÍDA DADOS}
fim
```

exercício 3.5 Permutar o conteúdo de duas variáveis na memória. O programa deverá iniciar preenchendo as duas variáveis por leitura e imprimindo os valores contidos nas variáveis. Em seguida, deve permutar o conteúdo das duas variáveis, ou seja, o conteúdo da primeira deve ficar na segunda e vice-versa. Imprimir novamente as duas variáveis para conferir se seus conteúdos foram realmente trocados.

Esta aplicação é muito comum e deve ser efetuada com cuidado. Ao solicitar a troca dos valores de duas variáveis na memória (`a` e `b`), pode-se pensar em fazer o seguinte:

```
a ← b
b ← a
```

Entretanto, ao colocar em `a` o valor contido em `b`, o valor que estava em `a` é perdido, conforme mostra a Figura 3.4. Para que isso não aconteça, o valor em `a` deve ser previamente guardado em uma variável auxiliar, para depois ser buscado para preencher a variável `b`, conforme ilustra a Figura 3.5. No algoritmo apresentado a seguir, as duas variáveis são preenchidas por leitura. Os valores nelas contidos são informados antes e depois da troca, para que se possa verificar se a operação foi realizada com sucesso.

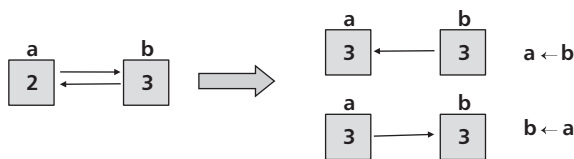



figura 3.4 Troca errada dos conteúdos de duas variáveis.

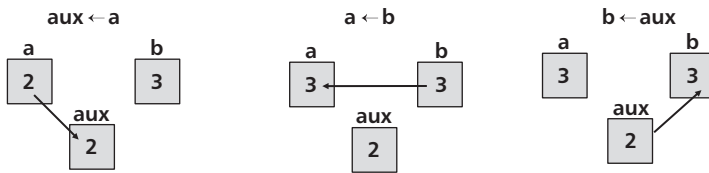



figura 3.5 Troca correta dos conteúdos de duas variáveis.

```
Algoritmo Permuta2Variáveis
{PERMUTA O CONTEÚDO DE DUAS VARIÁVEIS}
  Entradas: a, b (real)          {VARIÁVEIS A SEREM PERMUTADAS}
  Saídas: a, b                   {AS MESMAS VARIÁVEIS}
  Variável auxiliar: aux (real)
início
  ler (a, b)              {LÊ OS DOS DOIS VALORES}
  escrever (a, b)         {INFORMA OS VALORES LIDOS}
  aux ← a                 {PERMUTA O CONTEÚDO DAS DUAS VARIÁVEIS}
  a ← b
  b ← aux
  escrever (a, b)         {INFORMA VALORES TROCADOS}
fim
```

DICA DO PROFESSOR

Com a ferramenta VisuAlg, é possível desenvolver, testar e acompanhar a execução dos algoritmos, auxiliando no processo de ensino e aprendizagem de algoritmos.

A linguagem possui operadores lógicos, relacionais e aritméticos, comandos de entrada e saída e funções predefinidas para auxiliar no desenvolvimento dos algoritmos em pseudocódigo.

Assista ao vídeo para conhecer um pouco mais sobre essa ferramenta, compreender a estrutura básica e analisar algumas soluções práticas apresentadas de algoritmos sequenciais em forma de pseudocódigo, desenvolvida no VisuAlg.

Conteúdo interativo disponível na plataforma de ensino!

EXERCÍCIOS

1) **Observe o algoritmo:**

```
01 Algoritmo "calculo"
02 var
03      num1,num2,num3 , total: real
04 inicio
05      leia(num1,num2, num3)
06      total <- (quad (num1) + exp(num2,2) + raizq(num3) )
07      escreva("Resultado = ",total)
08 fimalgoritmo
```

Se, na linha de exibição dos dados, no comando de entrada "Leia", na linha 05, forem fornecidos os valores da tabela abaixo:

num1	2
num2	6
num3	4

Qual será o valor da variável "total" apresentado no comando de saída Escreva, na linha 07?

A) 46,0

B) 56,0

C) 55,2

D) 54,0

E) 42,0

2) **Considerando os operadores lógicos, relacionais e de atribuição utilizados na ferramenta de desenvolvimento de algoritmos em pseudocódigo VisuAlg, analise cada uma das seguintes afirmações e classifique em V (verdadeira) ou F (falsa).**

I – Os conectivos "e", "ou" e "não" são operadores lógicos.

II – O operador aritmético para realizar a divisão de inteiros é o símbolo "/"; para o resto da divisão, é Mod ou "^".

III – Os operadores relacionais utilizados são >, <, >=, <=, =, !=.

IV – O símbolo que representa uma atribuição é o “<-”.

A) V, V, F, F.

B) V, F, F, V.

C) F, V, F, V.

D) V, F, V, F.

E) V, V, V, V.

3) **Leia as coordenadas de dois pontos no plano cartesiano, calcule e imprima a distância entre esses dois pontos. A fórmula que calcula a distância entre os dois pontos (x1,y e (x2, y é dada por:**

$$\sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

Analise os algoritmos apresentados nas alternativas abaixo.

I

```
Algoritmo "Alternativa_I"
Var
  x1,x2,y1,y2,c1,c2,d: real
inicio
  Leia(x1,y1)
  Leia(x2,y2)
  c1 <- exp((x2 - x1),2)
  c2 <- exp((y2 - y1),2)
  d <- raizq(c1+c2)
  Escreva("Resultado = ",d)
Fimalgoritmo
```

II

```
Algoritmo "Alternativa_II"
Var
  x1,x2,y1,y2,c1,c2,d: real
inicio
  Leia(x1,y1)
  Leia(x2,y2)
  c1 <- (x2 - x1) ^ 2
  c2 <- (y2 - y1) ^2
  d <- raizq(c1+c2)
  Escreva("Resultado = ",d)
Fimalgoritmo
```

III

```
Algoritmo "Alternativa_III"
Var
  x1,x2,y1,y2, d: real
inicio
  Leia(x1,y1)
  Leia(x2,y2)
  d <- raizq(quad(x2-x1)+quad(y2-y1))
  Escreva("Resultado = ",d)
Fimalgoritmo
```

Quais alternativas apresentadas representam uma solução para o problema do cálculo da

distância entre dois pontos?

A) I.

B) I e II.

C) I e III.

D) II e III.

E) I, II e III.

4) **São dados três valores que representam as três notas de um aluno na disciplina de Algoritmos; os valores são representados por n1, n2 e n3. Calcule e imprima a média harmônica.**

Sabe-se que a média harmônica entre números reais positivos x1, x2, ..., xn é definida como sendo o inverso da média aritmética dos seus inversos, ou é o número de termos dividido pela soma do inverso dos termos, como apresentado na fórmula:

$$h = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} \dots + \frac{1}{x_n}}$$

Observe:

x1, x2.... xn: representam as notas n1, n2 e n3.

n: representa o número de termos, ou seja, a quantidade de notas.

Selecione a alternativa que contempla corretamente o comando de atribuição para o cálculo da média harmônica em pseudocódigo.

A) h <- n / (1 + (n1 * n2 * n3))

B) h <- n1 + n2 + 1/n3

C) h <- n / (1/n1 + 1/n2 + 1/n3)

D) h <- n/n1 +n/n2 + n/n3

E) h <- n/(1/(n1 + n2 + n3))

5) Um pedreiro necessita de auxílio para o cálculo de conversão de uma medida recebida em metros para centímetros e milímetros. O valor deve ser informado em metros e exibido para o pedreiro em centímetros e milímetros. Analise as soluções apresentadas nas alternativas e selecione a que representa a solução correta para o problema.

A)
```
Algoritmo “um”
Var m, cm: real
inicio
 Leia(m)
 cm <- m*100
 mm <- m * 1000
 Escreva(cm, mm)
fimalgoritmo
```

B)
```
Algoritmo “dois”
Var m, cm, mm: real
   inicio

   Leia(m)
   cm <- m*100
   mm <- m *1000
   Escreva(cm, mm)
```

```
fimalgoritmo
```

C)

```
Algoritmo “tres”
Var m, cm, mm: inteiro
inicio
  Leia(m)
  cm <- m*100
   mm <- m * 1000
   Escreva(cm, mm)
fimalgoritmo
```

D)

```
Algoritmo “quatro”
Var m ,cm,mm: real
inicio
   cm <- m*100
   mm <- m * 1000
   Escreva(cm, mm)
fimalgoritmo
```

E)

```
Algoritmo “cinco”
Var m,cm,mm: real
inicio
  leia (m)
  cm <- m*1000
  mm <- m * 100
  Escreva(cm, mm)
fimalgoritmo
```

NA PRÁTICA

O trânsito está um “terror”!

Ouvimos essa expressão todos os dias!

A cada dia que passa, mais veículos estão trafegando nas ruas das nossas cidades.

Vamos criar uma aplicação que calcula a despesa diária de um automóvel, para que possamos determinar o valor que podemos economizar com o transporte solidário, o qual, além das questões financeiras, apresenta muitas outras vantagens, como redução das emissões de carbono e diminuição dos congestionamentos.

Quais os dados de entrada necessários?

- Total de quilômetros percorridos no dia (quilômetros);

- custo por litro de combustível (preco_litro);

- média de quilômetros por litro que o veículo faz (media);

- total de gasto em estacionamento diário (estacionamento);

- total diário de gasto em pedágios (pedagio).

Qual a sequência correta do processamento?

- Calcular quantos litros gastamos no dia

litros <- quilometros / media

- Calcular o valor dos litros consumidos

Valor <- litros * preco_litro

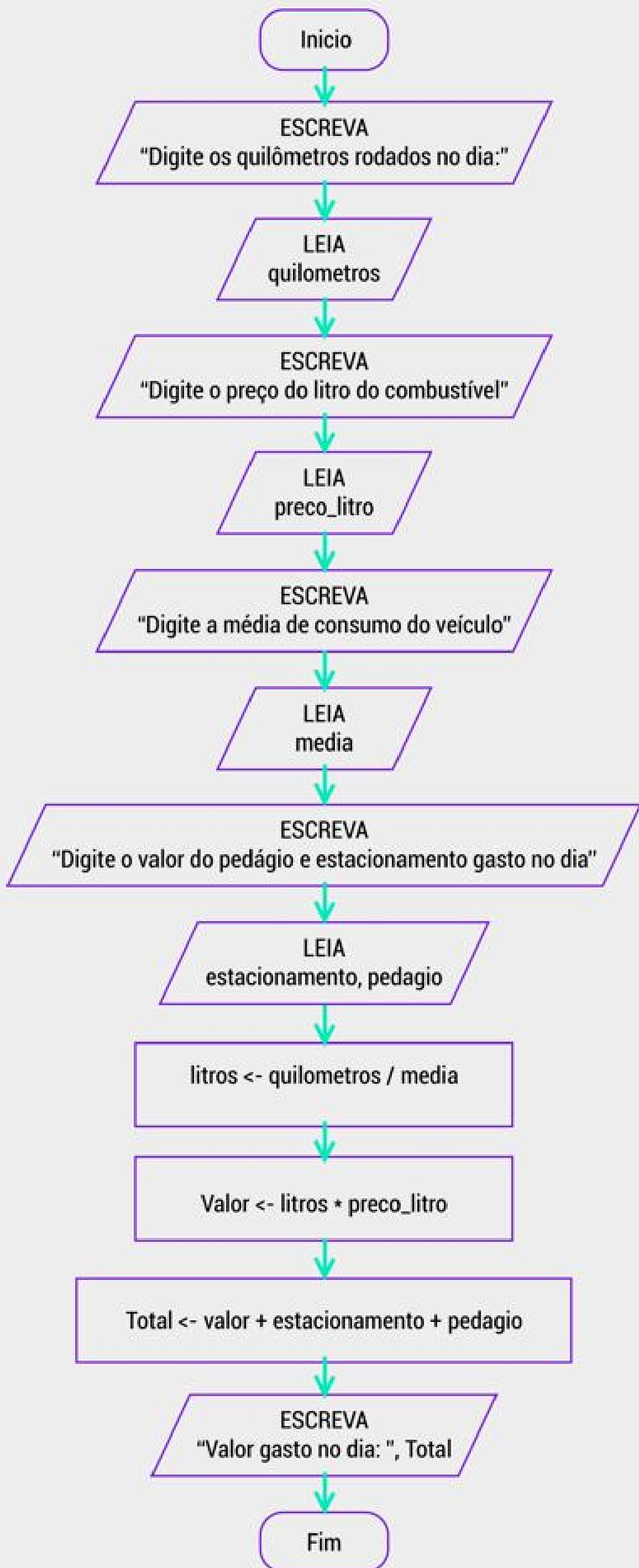
- Somar todas as despesas

Total <- valor + estacionamento + pedagio

Quais os dados de saída?

- O valor total em R$ gasto no dia.

Apresentamos, na tabela abaixo, a solução do problema em fluxograma e em pseudocódigo, para que possam ser comparados os comandos de entrada, saída e atribuição à solução do problema nas duas formas de representação.

SOLUÇÃO EM FLUXOGRAMA

SAIBA MAIS

Para ampliar o seu conhecimento a respeito desse assunto, veja abaixo as sugestões do professor:

Algoritmos e Programação: Primeiro Programa (VisuAlg)

Conteúdo interativo disponível na plataforma de ensino!

Lógica com VisuAlg 3.0 - Programação Sequencial

Conteúdo interativo disponível na plataforma de ensino!

Funções Predefinidas pelo VisualG

Conteúdo interativo disponível na plataforma de ensino!

A linguagem de programação do VisuAlg

Conteúdo interativo disponível na plataforma de ensino!

A tela principal do VisuAlg

Conteúdo interativo disponível na plataforma de ensino!

Menu do VisuAlg

Conteúdo interativo disponível na plataforma de ensino!